**Gesture Recognition**

The following table presents the results of different models experimented with to predict the gestures from the provided video data set.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Model Details** | **Result** | **Decision + Explanation** |
| **Architecture I: Conv3D** |  |  |  |
| **Model 1**  **(3D Convolution + Maxpooling + Flatten**  **)** | **Conv3D**  Total params: 25,749,637  Trainable params: 25,749,637 | **Train Accuracy: 0.19**  **Validation accuracy: 0.17** | **The model does not appear to learn per se. We proceed with incorporating additional 3D convolution layers in order to improve the accuracy.** |
| **Model 2**  **(Increased number of layers of 3D Convolution + Maxpooling + Flatten)** | **Conv3D** | **Generator error encountered** | **We reduce the pool size to address the error.** |
| **Model 3**  **(Increased number of layers of 3D Convolution + Maxpooling with lower spatial dimension + Flatten)** | **Conv3D**  Total params: 16,700,421  Trainable params: 16,700,421 | **Train Accuracy: 0.21**  **Validation Accuracy: 0.21** | **Accuracy is still considerably low. We proceed with incorporating batch normalization to improve accuracy.** |
| **Model 4**  **(Increased number of layers of 3D Convolution + Maxpooling with lower spatial dimension + Flatten + Batch normalization)** | **Conv3D**  Total params: 16,706,309  Trainable params: 16,703,365 | **Train Accuracy: 0.96**  **Validation Accuracy: 0.50** | **Model overfits. We add dropout to handle overfitting.** |
| **Model 5**  **(Increased number of layers of 3D Convolution + Maxpooling with lower spatial dimension + Flatten + Dropouts + batch normalization)** | **Conv3D**  Total params: 67,034,885  Trainable params: 67,032,965 | **Train Accuracy: 0.97**  **Validation Accuracy: 0.52** | **Overfitting is not addressed. We increase the drop out value from 0.2 to 0.5.** |
| **Model 6**  **(Increased number of layers of 3D Convolution + Maxpooling with lower spatial dimension + Flatten + Dropouts with 50% active neurons + batch normalization)** | **Conv3D**  Total params: 16,706,309  Trainable params: 16,703,365 | **Train Accuracy: 0.93**  **Validation Accuracy: 0.24** | **Since the validation accuracy reduced, we retain the dropout to 0.2. Next, we use Global Average Pooling instead of Flatten Layer.** |
| **Model 7**  **(Increased number of layers of 3D Convolution + Maxpooling with lower spatial dimension + GlobalAveragePooling3D + Droupouts + Batch normalization)** | **Conv3D**  Total params: 712,453  Trainable params: 710,533 | **Train accuracy: 0.79**  **Validation Accuracy: 0.73** | **Since the train and validation accuracies are now comparable, we believe that overfitting has been addressed. Next, we try to improve the accuracy further using TimeDistributed**  **layer.** |
| **Architecture II: Conv2D + RNN** |  |  |  |
| **Model 1** | **Conv2D**  Total params: 129,477  Trainable params: 128,517 | **Train accuracy: 0.93**  **Validation Accuracy: 0.53** | **Again, overfitting is observed. Next, we use the LSTM2D model.** |
| **Model 2** | **ConvLSTM2D**  Total params: 13,781  Trainable params: 13,589 | **Train accuracy: 0.82**  **Validation Accuracy: 0.52** | **The model is clearly overfitting. We do not proceed any further with the second architecture.** |

#### **We use Model 7 with Architecture I as the final model.**

#### We use**model-00029-0.44855-0.81250-0.65336-0.76562.h5**(obtained from the 29th epoch (of 30)) as our final model with the following performance metrics:

#### **training loss: 0.4486**

#### **validation loss: 0.6534**

#### **training accuracy: 0.8125**

#### **validation accuracy: 0.7656**